



Smooth Velocity

Программа сглаживания скоростной модели

версия 5.3.14

Описание

Tsunami Development

tsunamidevelopment.com

+1-713-783-1435

Tsunami Development

713-783-1435 tsunamidevelopment.com

The Information contained in this document is subject to change without notice. Tsunami Development assumes no responsibility for any error that may appear in this manual. Tsunami Development does not warrant that this document is error free. Please report any errors in this document in writing to Tsunami Development. Some states or jurisdictions do not allow disclaimer of expressed or implied warranties in certain transactions; therefore this statement may not apply to you.

СОДЕРЖАНИЕ

Введение	4
Инсталляция программы	6
Системные требования	6
Установка Tsunami	6
Установка графического интерфейса GUI	8
Плавающая лицензия	9
Конфигурация кластера	10
Утилита нодов	11
Начало работы	15
СОЗДАНИЕ ФАЙЛА СГЛАЖЕННЫХ СКОРОСТЕЙ	16
Закладка File Settings:	16
Закладка Velocity File	18
Закладка Options	20
Остановка и перезапуск задания	21
Приложение А: Информация лог файла	24
Лог-файл программы	24
Проверка лог-файла лицензии	27
Пример лог файла очистки нодов:	28
Приложение В: Информация файла параметров	29
Необходимые параметры	29
Список параметров в алфавитном порядке	29
Пример файла параметров	30
Приложение С: Инсталляция Java	31

ВВЕДЕНИЕ

Программа Raytracer – использует лучевое трассирование для создания времен пробега (traveltimes) для глубинной миграции до суммирования (PSDM). В случае если скоростная модель для трассирования недостаточно сглажена, лучи рассеиваются и тем самым создают переменчивые и неточные времена пробега. Сглаживание модели происходит и во время работы Raytracer, но пользователь не имеет возможности посмотреть насколько скорости сглаживаются и какие скорости используются. Для этого ему необходимо остановить программу, изменить входные параметры и перезапустить программу. Также, поскольку пользователь не применял сглаженную модель для расчета времен пробега, эта модель не может быть использована для последующей Томографии или в других программах обновления скоростной модели. В виду этих причин мы и решили создать отдельную программу, позволяющую пользователю разумно сглаживать скоростную модель и проверять её качество.

Процесс сглаживания модели не накапливает ошибку с глубиной. Этот процесс осредняет скорости таким образом, что они перемежаются между ускорениями и замедлениями, но в среднем остаются такими же. В тех местах, где градиент скорости высокий – сглаживание может чуть сдвинуть отражения, но эти сдвиги опять же будут локальными. В лог файле программы сглаживания приводится таблица со средними значениями сдвигов по глубине по отношению к двадцати одинаково расположенным интервалам глубины. Даже для моделей с высоким градиентом скорости, сглаживание в допустимых пределах не превышает одного-двух дискретов по глубине. Также сглаживание скоростной модели повышает производительность последующей программы лучевого трассирования.

Для того чтобы определить, насколько сглажена модель, мы рассчитываем максимальные изменения производных скорости в плоскостях x, y и z (dx, dy, dz). Чем больше сглажена модель, тем меньше будут значения этих производных. Для программы сглаживания, пользователь сам определяет необходимые значения для выходной сглаженной скоростной модели. Программа сама автоматически увеличит значения оператора сглаживания до тех пор, пока производные не станут меньше, чем необходимые. Таким образом, пользователь получает нужные значения производных скорости без необходимости угадывать фактор сглаживания.

Основываясь на опыте, мы рекомендуем задавать значения производных dx и dy не больше 15. Для dz , величина производной не должна превышать 50. В программе эти числа заданы по умолчанию.

Сглаженная скоростная модель может быть записана в формате SEG Y для последующего контроля качества, либо как блок-модель, или же в обоих форматах. Запись данных в формате блок-модели необходима для программы лучевого трассирования. Также блок-модель может быть создана в процессе работы программы трассирования, но если вы будете применять программу сглаживания скоростной модели, то эффективней будет создать её в процессе сглаживания. Обратите внимание, что в этом случае в программе трассирования опция создания новой блок модели должна быть выключена. Для более детального описания создания блок модели – смотрите главу “Создание сглаженной скоростной модели” и главу “Создание блок файла скоростной модели” в описании программы лучевого трассирования.

Программа сглаживания позволяет передискретизировать скоростную модель по глубине, что улучшает сглаживание и сводит к минимуму сдвиг горизонтов. Фактор делит дискрет по глубине на заданное пользователем число. Например, фактор 2 делит значение дискрета по глубине на половину, в результате мы получаем в два раза больше дискретов. Обратная сторона медали заключается в том, что наша модель увеличивается в объеме по сравнению с исходной. Принимая во внимание сей факт, необходимо избегать больших делителей. Также при больших числах время счета задания увеличивается, но опять же шаг дискретизации Скоростной модели никоим образом не влияет на время счета глубинной миграции, поскольку в ней используются таблицы времен пробега лучей, а не скоростная модель или блок файл скоростной модели.

ИНСТАЛЛЯЦИЯ ПРОГРАММЫ

Системные требования

Пакет Tsunami устанавливается на любых системах или их комбинациях: Linux, SGI, Opteron, Solaris и Itanium. Linux должен быть версии 2.4 или выше, с компилятором gcc 3.2, SGI IRIX версии 6.2 или выше, а также Itanium должен иметь Intel v8.0 компилятор.

У Tsunami удобная система инсталляции, которая позволяет соединять в один кластер различные версии операционных систем. Более того, ноды могут работать под различными версиями Linux. Пожалуйста, обратитесь в Tsunami Development, если вам необходимо соединить несколько версий Linux ОС.

Для расчета PSTM необходимо всего 256 MBytes памяти на каждый процессор, для расчета PSDM и Raytracer - 512 Mbytes. Меньший объем памяти может повлиять на производительность и время счета.

На вычислительных нодах не требуется дисковое пространство. Жесткий диск с данными подключается только к главному ноду. На вычислительных нодах должна существовать и быть доступной папка */tmp*. Во время работы, Tsunami мастер нод рассылает выполняемые файлы на каждый нод, и там же создаются локальные лог файлы.

Для работы Tsunami рекомендуется не использовать NFS систему на кластере. С увеличением количества процессоров до 100, NFS вызывает зависание процессов.

Команды *rsh* и *rcp* должны быть включены. Tsunami использует их для запуска процессов на вычислительных нодах. Поэтому пользователю необходимо открыть допуск и разрешение. Протестировать эти команды можно, набрав в командной строке следующее:

```
>rsh node_name date
```

```
>rcp file_name node_name:/tmp
```

Для большинства инсталляций достаточно 100 Mbit сети. Каждый нод должен быть подсоединен к 100 Mbit каналу. Этого вполне достаточно для работы Tsunami.

Установка Tsunami

1. Загрузите последнюю версию Tsunami, в формате tar с веб-сайта Tsunami Development.
2. Перейдите в директорию apps

```
>cd /apps
```
3. Создайте директорию tsunami

```
>mkdir tsunami
```
4. Скопируйте tar файл в директорию tsunami, созданную в п.3.
5. Раскройте tar файл:

```
>tar -xvpf tsunami.tar
```

У вас должны появиться следующие модули:

```
apps/tsunami/tsunami_5.2.13/pstm_5.2.13
                                psdm_5.2.13
                                rays_5.2.13
                                tomo_5.2.13
```

В директории каждого модуля содержится файл `readme`, а также директории Linux, SGI, Opteron, Solaris and Itanium с программными файлами. Лицензия контролирует, какие именно выполняемые файлы вы будете использовать. Конечная директория должна выглядеть следующим образом:

```
/apps/tsunami/tsunami_5.2.13/rays_5.2.13/itanium
                                itanium_3.5
                                linux
                                linux_3.4
                                opteron
                                opteron32
                                opteron_3.4
                                sgi
                                solaris
```

6. По электронной почте получите файл с лицензией от Tsunami Development. Сохраните его в директории TSUNAMI, созданной в п. 3.

Для инсталляции на SGI: Нет необходимости в дополнительных действиях.

Для инсталляции на Linux: Для Linux инсталляции, возможно, потребуется подключить программные файлы с компилятором `gcc`, чтобы они соответствовали текущим библиотекам Linux. Tsunami подсоединена к компилятору `gcc v3.2`. Вы можете проверить версию, набрав в командной строке `gcc -v`. Если у вас отличная от 3.2 версия, вам необходимо сделать следующее:

- a) Стереть файл `rays_linux` из директории
`/apps/tsunami/tsunami_5.2.13/rays_5.2.13/linux`

```
>rm /apps/tsunami/tsunami_5.2.13/rays_5.2.13/linux/rays_linux
```

- b) `>cd /apps/tsunami/tsunami_5.2.13/rays_5.2.13/linux/rays_objs`

- c) `>make -f makefile_rays_linux`

Эта команда соединит программные файлы с библиотеками и поместит их в папку
`/apps/tsunami/tsunami_5.2.13/rays_5.2.13/linux`.

- d) Повторите для PSDM and PSTM при необходимости.

Для инсталляции на Itanium: Intel v8.0 компилятор необходим для того, чтобы

системные библиотеки были доступны программе. Мы рекомендуем поместить компилятор в директорию, доступную для всех вычислительных узлов, так чтобы все они могли иметь доступ к библиотекам. Также компилятор Intel v8.0 может быть установлен на каждый узел. После того, как вы проверили наличие компилятора, добавьте в ваш “.cshrc” или “.bashrc” файл следующие строки:

```
Для .bashrc: export PATH="/opt/intel_cc_80/bin:$PATH"  
             source /opt/intel_cc_80/iccvars.sh
```

Для .cshrc: к списку переменной PATH добавьте /opt/intel_cc_80/bin.

Установка графического интерфейса GUI

1. Отдельной инсталляции для GUI не требуется. Для того чтобы запустить программу GUI (/apps/tsunami/tsunami_5.2.13/tsunami.jar), необходима Java 2 Platform, Standard Edition (J2SE) Version 1.4.2 или выше. Чтобы определить версию java на вашей системе наберите:

a) >java -version

Покажет версию java в вашей системе.

b) >which java

Покажет путь и директорию, где инсталлирована java.

Если у вас не инсталлирована java, или у вас имеется более ранняя версия, пожалуйста, обратитесь к вашему системному администратору. Он должен установить необходимую версию и прописать домашнюю директорию JAVA_HOME в .cshrc или .bashrc файл. Ниже, в дополнении C, приведена инструкция по установке Java .

2. Отредактируйте “.cshrc” или “.bashrc” файлы, включив следующие переменные:

a) Добавьте переменную JAVA_HOME, если таковой не существует.

Для .bashrc: export JAVA_HOME="java directory" Где java directory –
директория, куда инсталлирована java.

Для .cshrc: setenv JAVA_HOME java directory где java directory –
директория, куда инсталлирована java.

b) Добавьте JAVA_HOME/bin в переменные PATH.

Для .bashrc: PATH находится в файле .bash_profile.
Поместите JAVA_HOME/bin в конце
существующего списка переменной PATH.

Например: `export PATH=$PATH:$HOME/bin:$JAVA_HOME/bin`

Для `.cshrc`: `PATH` находится в файле `.login`

Поместите `JAVA_HOME_directory/bin` в конце существующего списка после переменной `set path=`.

Например: `set path=(/bin /usr/bin /sbin /usr/etc /usr/local/bin /usr/java2/bin)`

Где `/usr/java2` является домашней директорией `JAVA_HOME`

- c) Добавьте переменную `TSUNAMI` которая указывает на домашнюю директорию `tsunami`.

Для `.bashrc`: `export TSUNAMI="/apps/tsunami"`

Для `.cshrc`: `setenv TSUNAMI /apps/tsunami`

- d) Создайте `alias`, указывающий на выполняемый файл: .

Для `.bashrc`: `alias tsunami="/apps/tsunami/tsunami_5.2.13/tsunami.sh"`

Для `.cshrc`: `alias tsunami /apps/tsunami/tsunami_5.2.13/tsunami.sh`

Плавающая лицензия

Лицензия контролирует количество работающих нодов. До версии 3.1.7 пользователь мог запускать программу на всех или любых, выбранных им и лицензированных нодах. При использовании плавающей лицензии, пользователь может перечислить в файле все имеющиеся у него ноды, но работать будет только то количество, на которое приобретена лицензия.

Начиная с версии 4.15.9, количество процессоров также учитывается, для случаев, где нод имеет больше чем 2 процессора.

Программа проверит лицензию при запуске задания на счет, а также по его окончании или остановке с последующим перезапуском. Любое количество заданий может считаться на нодах, но только на том количестве нодов, которое лицензировано. Например, из 50 нодов, лицензированы только 20. При этом пользователь может использовать любые 20 из 50 одновременно.

При использовании плавающей лицензии для процессоров, во время счета только одного задания на ноде, лицензия проверит количество процессоров, указанных в файле `node.db`. Если на ноде считается несколько заданий, то все лицензированные процессоры будут проверены и в лог файле появится предупреждающее сообщение.

В случае если не все процессоры на ноде лицензированы, то пользователь не сможет запустить несколько параллельных заданий. Программа проверяет количество процессоров на ноде, сверяет их с количеством, указанным в лицензии и с `node db` файлом.

Проверить статус лицензии можно при помощи утилиты: `check_license_file`, вызываемой из меню **Utilities**. В окошке появится сообщение об именах и количестве нодов, их MAC-адресах, дате истечения срока действия лицензии и количестве лицензированных процессоров

```
> check_license_file <license file name>
```

В случае если проверка лицензии по какой-то причин не сработала, то перезапустить лицензию можно при помощи утилиты `clear_nodes`. Эта программа убивает все процессы на нодах, перечисленных в `node.db` файле, а также переустанавливает лицензию на эти ноды. Утилита `clear_nodes` была изменена, начиная с версии 3.1.7, и больше не использует `user id` для определения и выбора процессов, идущих на нодах. Теперь, при запуске `clear_nodes`, все процессы идущие на нодах убиваются, при этом создается файл `clear_nodes.log` в домашней директории `$TSUNAMI`.

```
>clear_nodes <node db file>
```

Файлы лицензий помещаются в директорию `/apps/tsunami`.

Конфигурация кластера

Для работы Tsunami необходимо создать файл с именами нодов, `node.db`. Это ASCII файл следующего формата:

Node_name	number_of_processors	memory_in_Mbytes	speed_factor	operating_system
master04	1	1000	1.0	linux
node01	2	1000	1.0	linux
node02	2	1000	1.0	linux

Имена нодов указаны в файле `/etc/hosts`.

Показатель скорости, `speed factor`, отражает относительную скорость процессора по отношению к другим нодам. Показатель скорости и количество процессоров используются для распределения задания по нодам, позволяя одновременно подключать на один кластер ноды с различным количеством процессоров и их разной скоростью.

Обычно, быстрым процессорам задается значение 1.0, а более медленным – значение меньше 1.0. К примеру, 3000 MHz процессор имеет значение 1.0, при этом 1500 MHz процессор - значение 0.5.

По завершении задания, в конце лог файла приводится статистика для каждого нода. Используйте значения *“Millions of shift and sums per second kernel time”* как показатель скорости счета миграции на нодах.

Название операционной системы, указывает программе, какой из выполняемых файлов использовать для расчета задания: `linux`, `sgi`, `solaris`, `opteron` and `itanium` (`linux`).

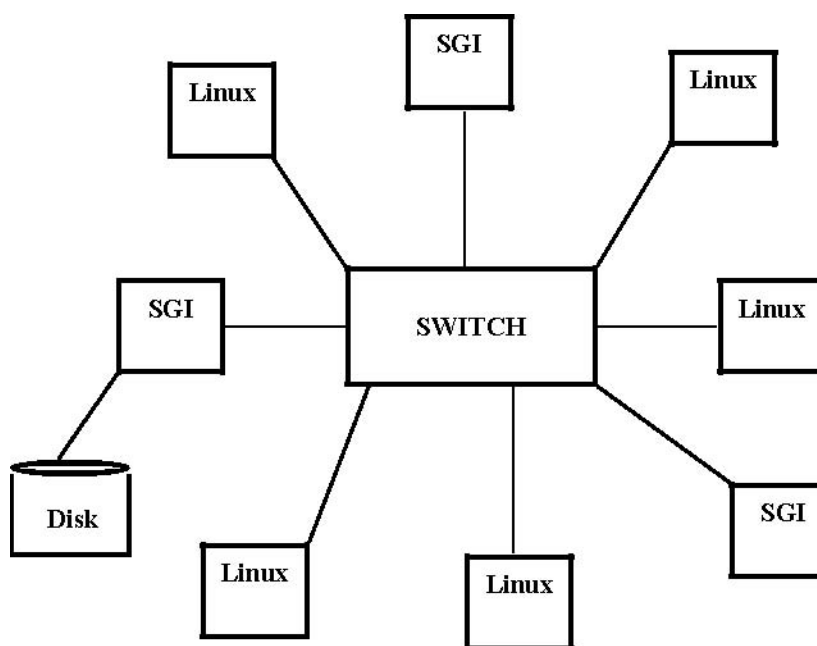
Главный нод (`master node`) обязательно прописывается в первой строке `node.db` файла. Это тот нод, с которого запускается программа, и только тот, который имеет доступ к входным данным и файлам для записи выходных данных. Он также может быть использован и как вычислительный нод. В этом случае, его необходимо указать еще раз, второй строкой.

Например , node db, такой конфигурации:

server1	2	512	1.0	linux
server1	2	512	1.0	linux
linux1	2	512	1.0	linux
linux2	2	512	1.0	linux
sgi1	4	512	.75	sgi
itanium1	8	1000	.65	Itanium

Для того чтобы программа работала, должна быть доступна rsh опция с главного нода на все остальные вычислительные ноды. Если у вас возникнут вопросы, обратитесь к вашему системному администратору или к ESS. Директория /tmp должна быть доступна на всех нодах. В неё записываются выполняемые файлы, при запуске задания, которые затем удаляются после его завершения.

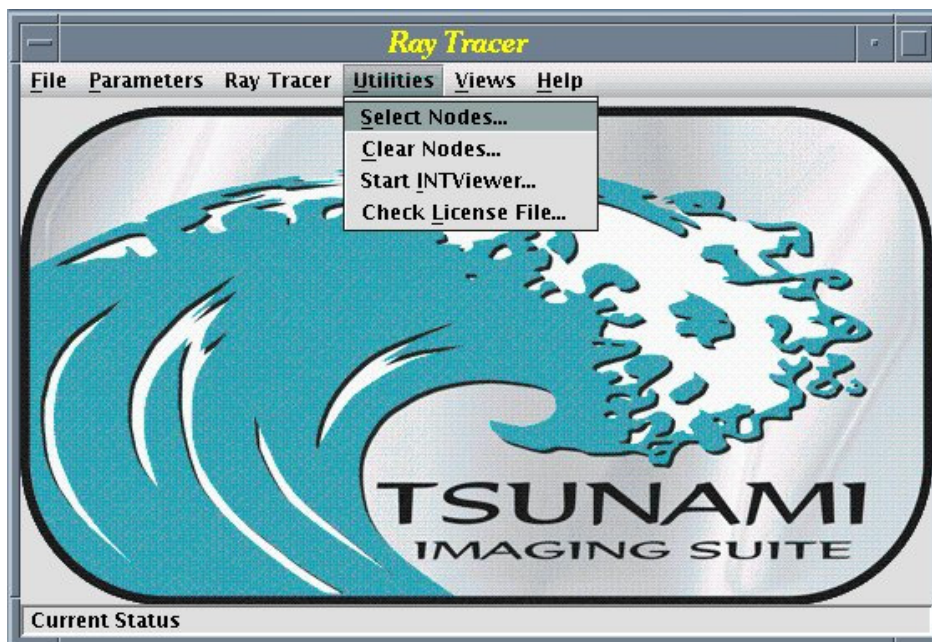
Пример конфигурации кластера



Утилита нодов

После того, как вы создали node.db файл, вы можете использовать опцию **Select Nodes** из главного меню GUI. *Смотри: **Начало работы с графическим интерфейсом (GUI)**.

1. В заголовке **Utility** —> выберите **Select Nodes**.



Нажав на **Select Nodes**, пользователь открывает окно с дополнительным меню с возможностью выбора нодов, а также создания новых файлов.

Для того чтобы открыть файл со списком нодов, или создать новый файл, программа должна найти master node файл. Он находится в домашней директории \$TSUNAMI, например *"/apps/tsunami"*. Если он не найден, то пользователю предлагается указать директорию, в которой он находится.

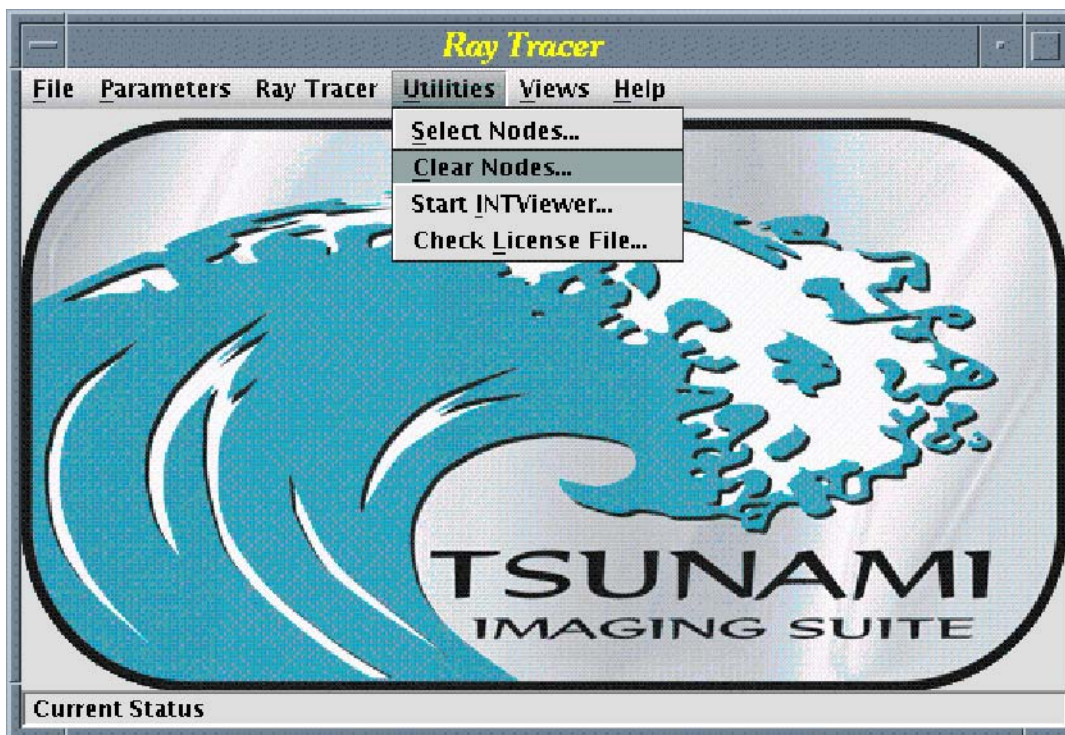
Если master node файл найден и указана директория текущего проекта (на закладке Processing), откроется диалоговое окошко с заданными нодами. Пользователь также может выбрать нужные ему ноды и сохранить их, как новый файл для данного задания.

Если пользователь сохраняет данный файл со списком нодов, то этот файл автоматически сохраняется в файл параметров, как параметр **nddb=**. (Находится в закладке Processing).

Для более детальной информации по созданию файла со списком нодов, а также распределению задания по нодам смотрите раздел **Анализ производительности и улучшенные возможности**.



2. Если задание было остановлено или прекратилось считаться по причине проблем с “железом“, программа Tsunami сразу же удаляет все процессы на вычислительных нодах. В некоторых случаях необходимо запустить утилиту Clear Nodes. Эта программа убивает все процессы на нодах, перечисленных в node.db файле, кроме мастер нода, а также переустанавливает лицензию на эти ноды. Лог файл будет помещен в домашнюю директорию *\$TSUNAMI*.



Clear Nodes с командной строки:

В командной строке наберите:

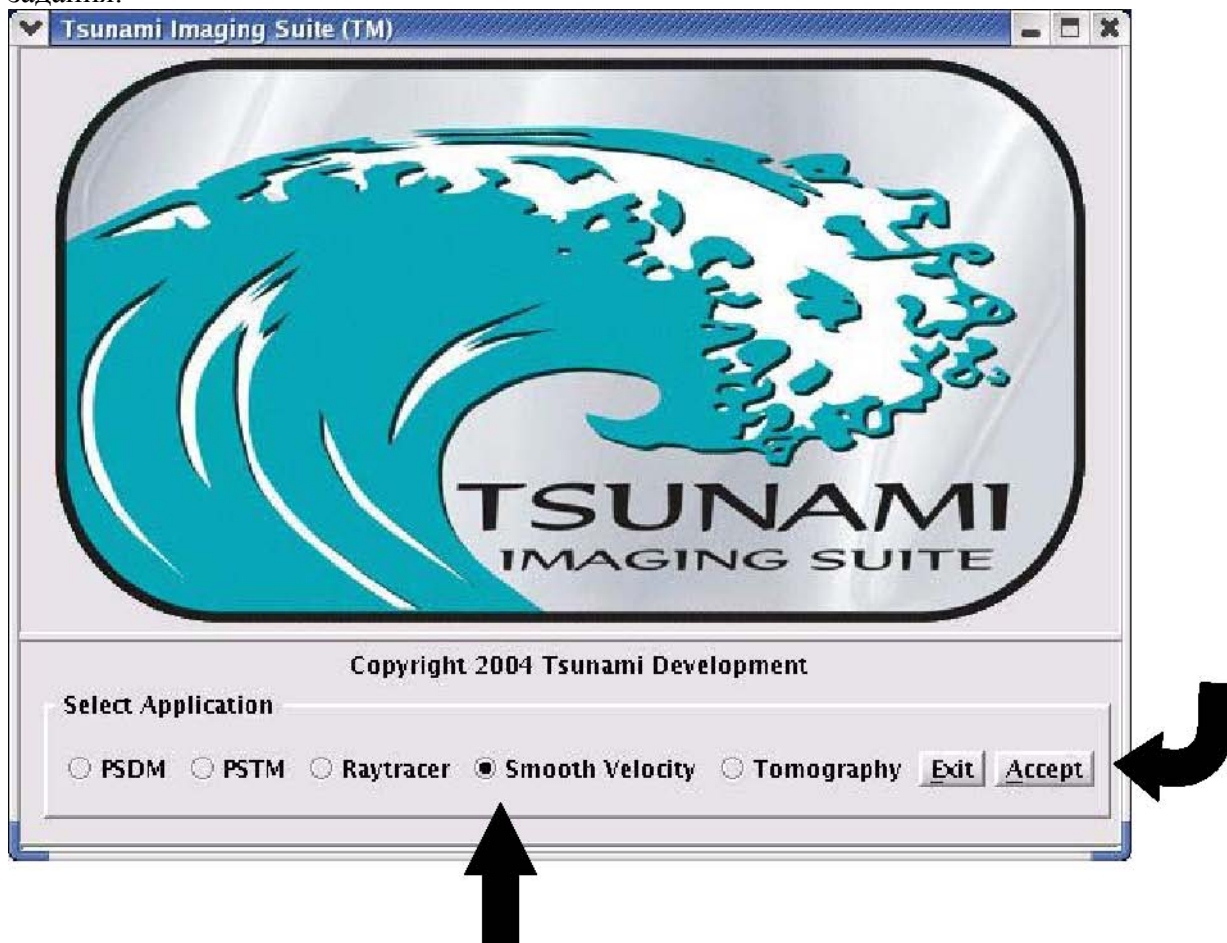
```
> clear_nodes <node.db file>
```

Эта программа убивает все процессы на нодах, перечисленных в node.db файле, кроме мастер нода, а также переустанавливает лицензию на эти ноды. Процессы, идущие на мастер ноде удалятся, как результат удаления процессов на вычислительных нодах. Для дополнительной информации см. раздел Плавающая лицензия.

НАЧАЛО РАБОТЫ

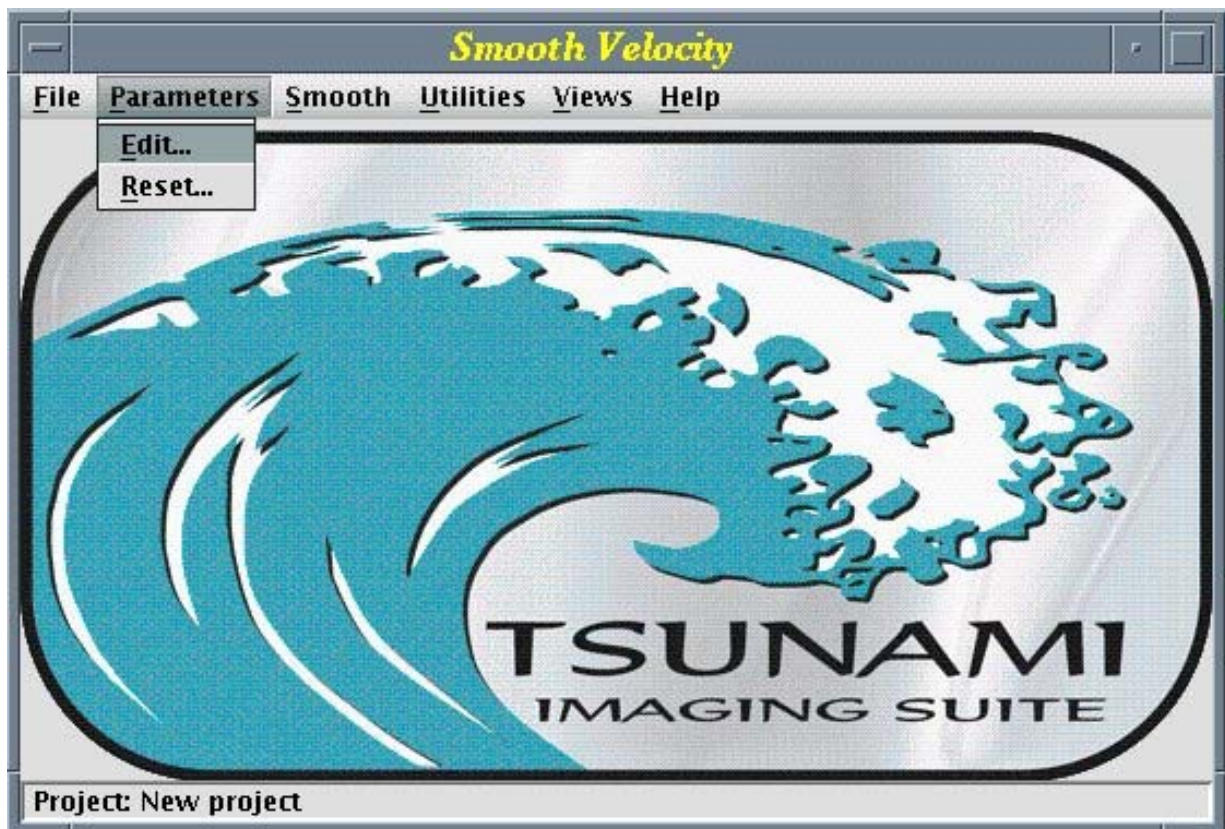
Используйте alias , созданный в п.2 раздела инсталляция GUI для запуска программы: Пример
<prompt>: tsunami

- 1 Выберите кнопку Smooth Velocity из окошка Startup и нажмите Ассепт.
- 2 Откройте новый файл, выберите File —> New для создания параметров нового задания.



СОЗДАНИЕ ФАЙЛА СГЛАЖЕННЫХ СКОРОСТЕЙ

- 1 В окошке Smooth Velocity выберите **Parameter** → **Edit**.
- 2 Откройте закладку **Files** в окне выбора параметров **Parameter Selection Box**. Все требуемые для заполнения поля параметров выделены жёлтым цветом.



Закладка File Settings:

Application Directory (hmdr): Указывает путь к выбранной вами версии Tsunami Smooth Velocity. Например: Для версии 5.2.13 она должна показывать на ~/tsunami_5.2.13/smth_5.2.13

Project Directory (prjdr): Директория задания, в которую будут помещены лог файлы, а также временные файлы, создаваемые во время расчета задания. В этой же директории создается временный блок файл скоростей.

Log file name (logfile): Если вы не указываете имя лог файла вручную, то по умолчанию он создается с именем project_dir/jobname.log. Если вы указываете имя лог файла вручную, то рекомендуется указывать весь путь, включая директорию задания.

Input Interval Velocity File Name (velf): Указываете путь к кубу интервальных скоростей в формате SEG Y. Файл должен иметь регулярную дискретизацию по инлайнам, кросслайнам и глубине. Также файл должен быть отсортирован по

инлайн, и все инлайны должны содержать одинаковое количество кросслайнов, т.е. модель должна быть прямоугольной формы. Модель может быть в формате IEEE или IBM 32 floating point, и если IEEE – то считываются оба, как IEEE big так и IEEE little endian форматы.

Output SEG Y Velocity File (velfo): Задаете имя и указываете путь к файлу скоростей в формате SEG Y, создаваемому программой. Этот файл в дальнейшем используется для визуализации модели и контроля её качества. Используется для трассирования а также файл может быть использован для ввода в программу томографии.

Output Velocity Block File (blkf): Задаете имя и указываете путь к временному блок файлу скоростей, создаваемому программой. Этот блок файл используется программой лучевого трассирования. Для этого необходимо выключить опцию Build the New Block Model в задании параметров трассирования лучей на страничке Velocity File.

NB: Вы можете указать, какой из файлов (SEG Y или блок файл) создастся программой, или же можете указать создание обоих файлов. Если вы сглаживаете скорости для глубинной миграции, то следует создать SEG-Y, а блок-файл строить в программе RayTracer, используя сглаженный SEG-Y в качестве входного скоростного файла.

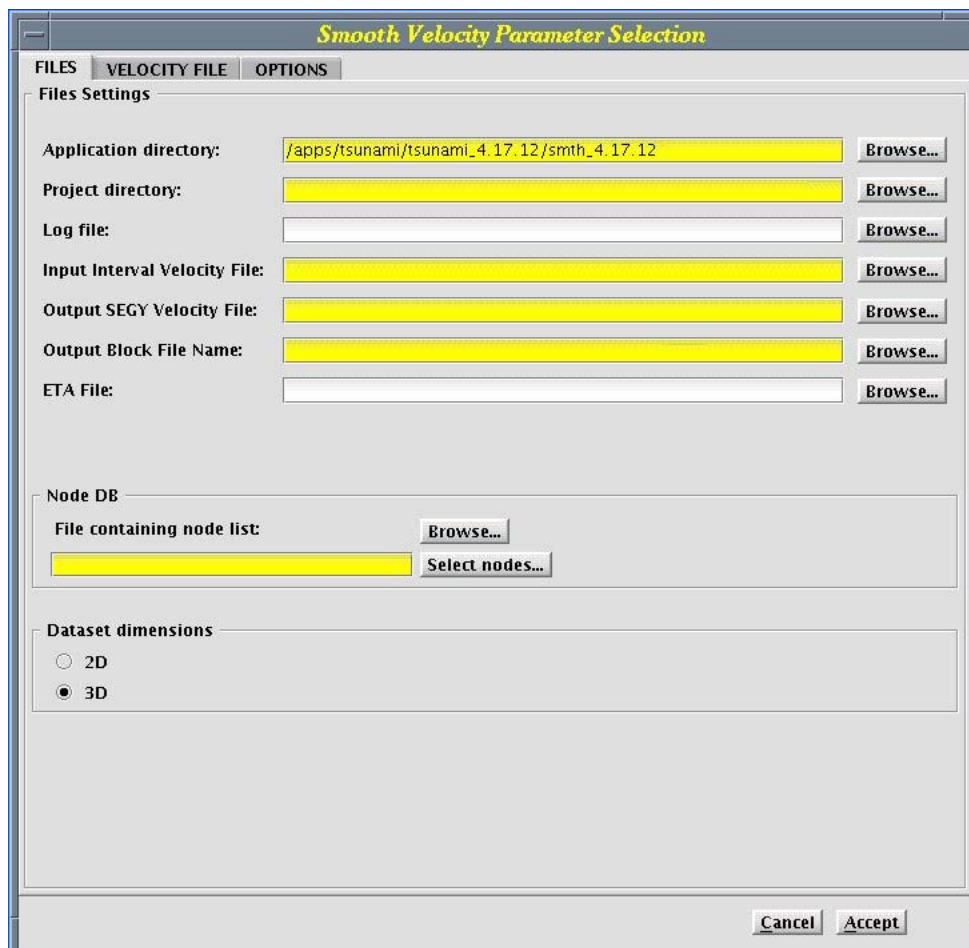
ETA File (etaf): Задаете имя и указываете путь к SEG Y файлу со значениями анизотропийного параметра «эта» (η). Файл должен иметь те же размеры и тот же шаг по инлайнам, кросслайнам и вертикали, что и скоростной файл. Положение требуемых параметров в заголовке файла «эта» должно быть идентичным заголовку скоростного файла.

Список нодов:

File containing node list (nddb): Список нодов, используемых для расчета задания. Дополнительную информацию по созданию списка см. в разделе Конфигурация кластера.

Browse: Открывает диалоговое окошко для выбора файла со списком нодов.

Select Nodes: Позволяет редактировать выбранный вами список нодов. В этом окошке можно добавить новые ноды, удалить имеющиеся и сделать доступными или недоступными ноды для конкретного задания.



Файл параметров для запуска программы с командной строки:

Закладка **Files Setting** в командной строке представляет собой следующий список параметров:

hmdir= домашняя директория программы
prjdr= директория задания (проекта)
logfile= имя лог файла
velf= куб скоростей в формате SEGY
blkf= блок файл скоростей, созданный программой
velfo= куб сглаженных скоростей в формате SEGY
nddb= файл со списком узлов
twod= 2D / 3D формат данных

Дополнительная информация по созданию файла параметров с командной строки и полный список требуемых параметров приводится в приложении В: Данные файла параметров.

Закладка **Velocity File**

3. Выберите закладку **Velocity File** в окне выбора параметров **Parameter Selection**. Введите необходимую информацию. Все требуемые для заполнения поля параметров выделены жёлтым цветом.

Header loc of Xline (vxlb): номер байта с номером кросслайна в заголовке скоростной модели (По умолчанию значения нет)

Header loc of inline (vilb): номер байта с номером инлайна в заголовке скоростной модели (По умолчанию значения нет)

Depth Increment (vdz): Шаг скоростной модели по глубине.

Velocity File Label (vxlilty): Формат значений в заголовках, по умолчанию 0 или 4 byte integer. 0 = 4 byte integer и 1= 2 byte integer.

Distance between xlines (xldist): Расстояние между кросслайнами (По умолчанию значения нет)

Distance between inlines (ildist): Расстояние между инлайнами

The screenshot shows a dialog box titled "Smooth Velocity Parameter Selection" with three tabs: "FILES", "VELOCITY FILE", and "OPTIONS". The "VELOCITY FILE" tab is active. The dialog is divided into several sections:

- Velocity File Settings:**
 - Inline/Xline Byte Location:** Two yellow input fields for "Inline Byte Location" and "Xline Byte Location".
 - Velocity file label:** Radio buttons for "2 bytes" (unselected) and "4 bytes" (selected).
 - Velocity Depth Increment:** A yellow input field.
- Distance:**
 - "Distance between inlines:" followed by a yellow input field.
 - "Distance between xlines:" followed by a yellow input field.
- Velocity Model Format:** Radio buttons for "IEEE Little Endian" (unselected), "IEEE Big Endian" (unselected), and "IBM" (selected).
- Velocity Model Header Format:** Radio buttons for "IEEE Little Endian" (unselected), "IEEE Big Endian" (selected), and "IBM" (unselected).
- Velocity Model Format Output:** Radio buttons for "IEEE Little Endian" (unselected), "IEEE Big Endian" (unselected), and "IBM" (selected).

At the bottom right, there are "Cancel" and "Accept" buttons.

Velocity Model Format (dataf): Показывает формат скоростной модели. Может быть выбран как исходный формат, так и IEEE_LE, IEEE_BE или IBM. (По умолчанию - IBM).

Velocity Model Header Format (hdrfmt): Показывает формат заголовка скоростной модели.. Может быть выбран как исходный формат, так и IEEE_LE, IEEE_BE или IBM. (По умолчанию - IBM).

Trace data format output (datafo): Показывает формат выходной скоростной модели. Может быть выбран как исходный формат, так и IEEE_LE, IEEE_BE или IBM.

(По умолчанию - исходный формат).

Файл параметров для запуска с командной строки:

Закладка **Velocity File** в командной строке представляет собой следующий список параметров:

vxlb= номер byte кросслайн в заголовке скоростной модели (По умолчанию значения нет)
vilb= номер byte инлайн в заголовке скоростной модели (По умолчанию значения нет)
vdz= шаг скоростной модели по глубине
xldist= Расстояние между кросслайнами (По умолчанию значения нет)
ildist= Расстояние между инлайнами (По умолчанию значения нет)
dataf= Показывает формат скоростной модели. Может быть выбран как исходный формат, так и IEEE_LE, IEEE_BE или IBM. (По умолчанию - IBM).
hdrfmt= Показывает формат заголовка скоростной модели.. Может быть выбран как исходный формат, так и IEEE_LE, IEEE_BE или IBM. (По умолчанию - IBM).
vxliltyp= Формат значений в заголовках, по умолчанию 0 или 4 byte integer.
о кросслайнам.
datafo= Показывает формат выходной скоростной модели. Может быть выбран как исходный формат, так и IEEE_LE, IEEE_BE или IBM. (По умолчанию - исходный формат).

Дополнительная информация по созданию файла параметров с командной строки и полный список требуемых параметров приводится в приложении В: Данные файла параметров.

Закладка *Options*

4. Закладка **Options**. Все требуемые для заполнения поля параметров выделены жёлтым цветом.

Smoothing Operator Controls: операторы сглаживания

Maximum Target dx (dxmx): Максимальное значение производной в плоскости X, полученное после сглаживания. По умолчанию = 15

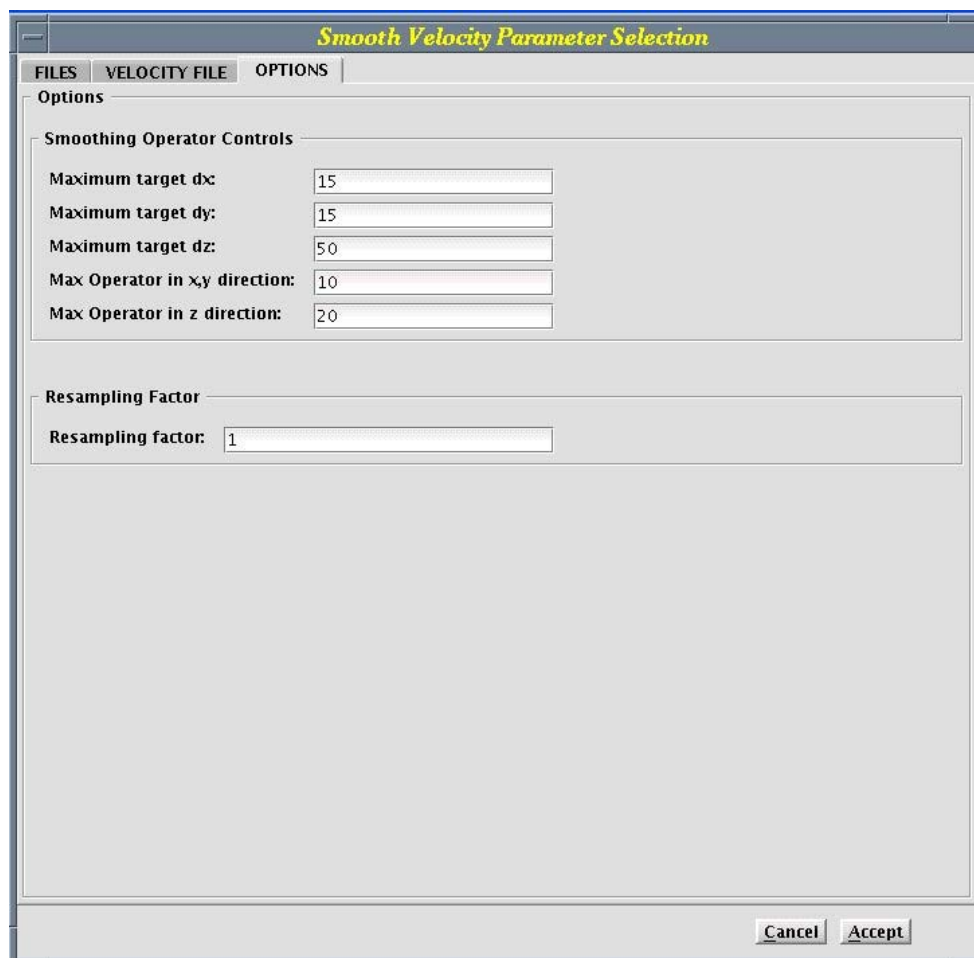
Maximum Target dy (dymx): Максимальное значение производной в плоскости Y, полученное после сглаживания. По умолчанию = 15

Maximum Target dz (dzmx): Максимальное значение производной в плоскости Z, полученное после сглаживания. По умолчанию = 50

Max Operator in dx, dy direction (mxopxy): По умолчанию 10

Max Operator in dz direction (mxopz): По умолчанию 20.

Resampling factor: Значение оператора прореживания. По умолчанию 1. Пример: фактор 2 создаст в два раза больше дискретов по глубине.

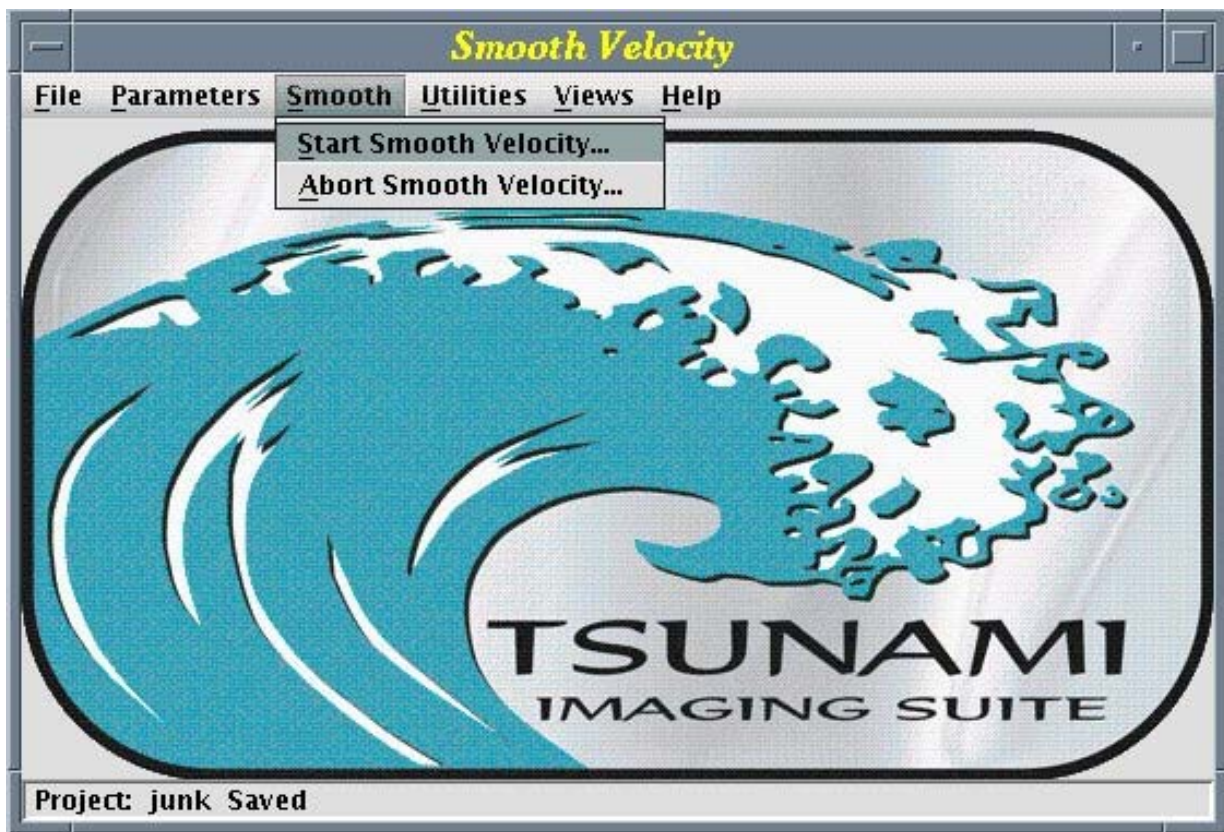


ОСТАНОВКА И ПЕРЕЗАПУСК ЗАДАНИЯ

Остановить/Перезапустить задание из GUI

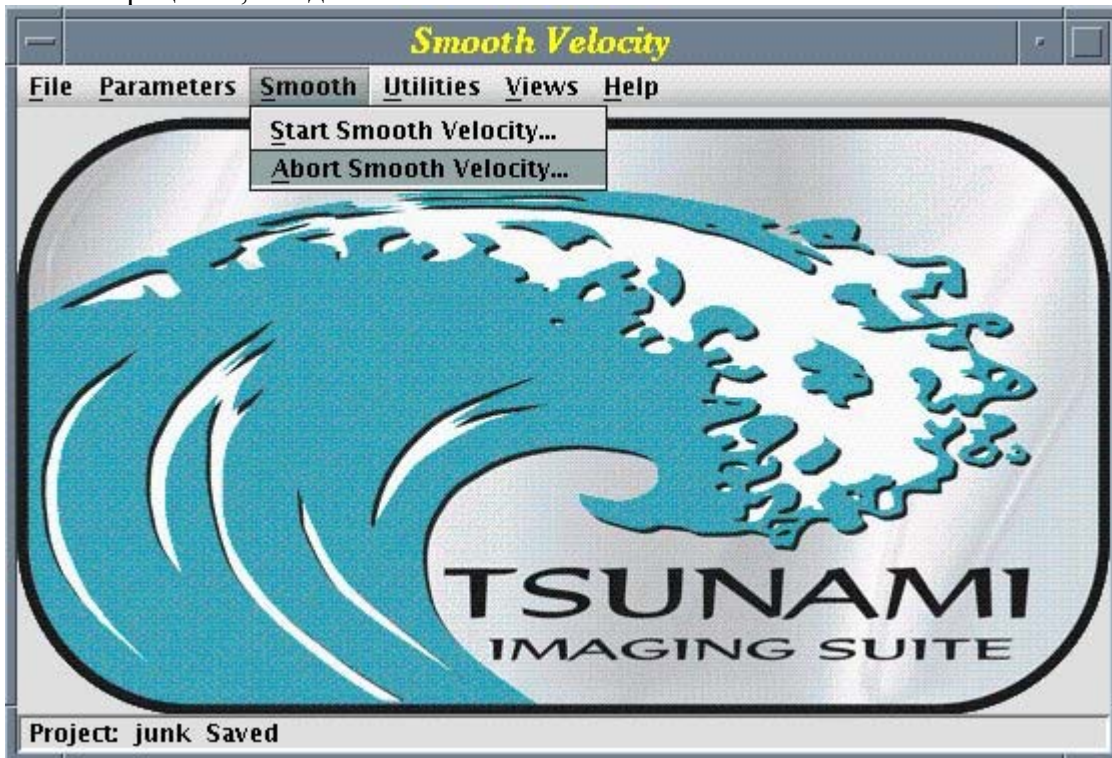
Start Smooth Velocity – Запуск лучевого трассирования

Параметр Start Smooth Velocity запускает задание с выбранным или созданным файлом параметров. Текущий файл параметров показан в нижней строке графического интерфейса. Как только задание начнет считаться, то появится сообщение : "Starting Smooth Velocity for file: (file name)."



Параметр Abort Smooth Velocity

Как только пользователь задает этот параметр, задание немедленно прерывается, на нодах удаляются все процессы, и задание полностью останавливается.



Остановка/Перезапуск задания из командной строки:

Для запуска задания наберите в командной строке следующее:

```
smth_start < parameter file name > &
```

Для полной остановки задания, наберите :

```
smth_abort < parameter file name >
```

Задание немедленно прервется, на нодах удалятся все процессы и задание полностью остановится.

ПРИЛОЖЕНИЕ А: ИНФОРМАЦИЯ ЛОГ ФАЙЛА

Лог-файл программы

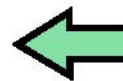
Лог-файл находится в директории проекта : ~proj_dir/parameter_file.log

Программа Smooth Velocity предоставляет много полезной информации в своем лог файле. При запуске нового задания на трассирование, проверьте и убедитесь, что вы правильно считываете данные скоростной модели.

Для удобства, при старте программы лог распечатает интервал значений инлайнов и кросслайнов, которые покрывает скоростная модель. Далее, будут распечатаны значения минимальной и максимальной скоростей в данной модели, а также данные заголовка первой трассы исходных данных. Эти значения должны быть проверены пользователем и сопоставлены с ожидаемыми.

Version 4.17 Tsunami Velocity Smoother
Parameter file: smth_29.param

```
hmdr= '/data3/software/tsunami_4.17.10/smith_4.17.10'  
velf= '/data1/summit/grid32_Vint_ibm.segy'  
velfo= '/data3/smith_test/summit/grid32_Vint_ibm_smth-1.segy'  
blkf= '/data3/smith_test/summit/smooth.blk  
nddb= '/data3/smith_test/summit/node.db'  
prjdr= '/data3/smith_test/summit  
ildist= 82.5  
xldist= 82.5  
resamp= 3  
vilb= 9  
vxlb= 17  
vdz= 32  
dxmx= 15  
dymx= 15  
dzmx= 50  
mxopxy= 20  
mxopz= 20vilb= 1
```



Проверка 1

vdz= , vilb=, vxlb=,
xldist= and ildist=

```
Jun 23 15:54  
Reading license file /home/software/psdm_license.dat  
Current license expires month 1 day 31 year 2006  
Reading the velocity model  
Checking the sort order, and the limits of model  
Read 150000 velocity traces  
Read 160000 velocity traces  
Read 170000 velocity traces
```



Проверка 2

Проверьте правильность
лицензионного файла

Read 180000 velocity traces
Read 190000 velocity traces
Read 200000 velocity traces
Velocity model coverage: First xline 1000 Last xline 1454 xline inc 1 First inline
Velocity model z increment: 32
Reading velocity model to create smoothing operator
First pass 5.0 percent complete
First pass 10.0 percent complete
First pass 15.0 percent complete
First pass 20.0 percent complete
First pass 25.0 percent complete
First pass 30.0 percent complete
First pass 35.0 percent complete
First pass 40.0 percent complete
First pass 45.0 percent complete
First pass 50.0 percent complete
First pass 55.0 percent complete
First pass 60.0 percent complete
First pass 65.0 percent complete
First pass 70.0 percent complete
First pass 75.0 percent complete
First pass 80.0 percent complete
First pass 85.0 percent complete
First pass 90.0 percent complete
First pass 95.0 percent complete
Pass one completed
Smoothing model
Outputting segy file
Outputting block file
Final pass 5.0 percent complete
Final pass 10.0 percent complete
Final pass 15.0 percent complete
Final pass 20.0 percent complete
Final pass 25.0 percent complete
Final pass 30.0 percent complete
Final pass 35.0 percent complete
Final pass 40.0 percent complete
Final pass 45.0 percent complete
Final pass 50.0 percent complete
Final pass 55.0 percent complete
Final pass 60.0 percent complete
Final pass 65.0 percent complete
Final pass 70.0 percent complete
Final pass 75.0 percent complete
Final pass 80.0 percent complete
Final pass 85.0 percent complete
Final pass 90.0 percent complete
Final pass 95.0 percent complete

Length of smoothing operator x 11 y 12 z 8
Minimum vel in model 5413.09
Maximum vel in model 14500.00
Initial dx 108.3 dy 108.3 dz 281.0
Final dx 14.9 dy 14.2 dz 46.9

Depth Ave Shift

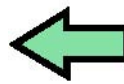
961.60 0.80
1923.20 1.65
2884.80 2.63
3846.40 3.75
4808.00 4.85
5769.60 5.79
6731.20 6.30
7692.80 6.85
8654.40 7.57
9616.00 8.19
10577.60 8.65
11539.20 9.14
12500.80 9.71
13462.40 10.94
14424.00 13.04
15385.60 15.32
16347.20 17.47
17308.80 19.45
18270.40 21.09
19232.00 22.71

Successful Completion: Smooth Velocities

Однообразные строчки стёрты для экономии места

Проверка лог-файла лицензии

Product name rays
Creation date 010104
Start date 010104
End date 123104
Length of hw record 96
Number of licensed nodes 100



Проверка 1
Даты начала и окончания
действия лицензии

Mac Address	Node Name	Checked Out
00:A0:C9:FB:25:B4	dual450	
00:42:52:00:13:E1	rlx-0-0-1	
00:42:52:00:0F:37	rlx-0-0-2	
00:42:52:00:13:36	rlx-0-0-3	1
00:42:52:00:11:38	rlx-0-0-4	
00:42:52:00:23:0B	rlx-0-0-5	
00:42:52:00:0B:EF	rlx-0-0-6	
00:42:52:00:18:FD	rlx-0-0-7	1
00:42:52:00:1C:3C	rlx-0-0-8	
00:42:52:00:19:27	rlx-0-0-9	
00:42:52:00:25:B4	rlx-0-0-11	
00:42:52:00:13:24	rlx-0-0-13	1
00:42:52:00:17:26	rlx-0-0-15	
00:42:52:00:1A:35	rlx-0-0-17	
00:42:52:00:1A:0B	rlx-0-0-19	
00:42:52:00:18:7C	rlx-0-0-21	
00:E0:81:03:38:C6	linux1	
00:03:47:71:D2:54	linux2	1
00:03:47:71:E9:F5	linux3	
00:03:47:6B:45:47	linux4	1
00:E0:81:02:B4:3E	linux5	
00:03:47:71:62:FC	linux6	
00:03:47:71:5B:73	linux7	
00:03:47:71:D2:41	linux8	
00:03:47:71:65:23	linux9	
00:03:47:71:62:2D	linux10	
00:03:47:71:5D:B9	linux11	
00:03:47:71:5D:BB	linux12	1
00:03:47:71:5B:7C	linux13	1
00:03:47:71:D2:49	linux14	1

}



Проверка 2
Единицка означает, что
нод отключён от лицензии

Пример лог файла очистки нодов:

```
clearing node rlx-0-0-1  
clearing node rlx-0-0-2  
clearing node rlx-0-0-3  
clearing node rlx-0-0-4  
clearing node rlx-0-0-5  
clearing node rlx-0-0-6  
clearing node rlx-0-0-8  
clearing node rlx-0-0-9  
clearing node rlx-0-0-11  
clearing node rlx-0-0-13  
clearing node rlx-0-0-15  
clearing node rlx-0-0-17  
clearing node rlx-0-0-19  
clearing node rlx-0-0-21  
Resetting license file /apps/tsunami/pstm_license.dat  
Reading license file /apps/tsunami/pstm_license.dat  
Resetting license file /apps/tsunami/psdm_license.dat  
Reading license file /apps/tsunami/psdm_license.dat
```

ПРИЛОЖЕНИЕ В: ИНФОРМАЦИЯ ФАЙЛА ПАРАМЕТРОВ

Необходимые параметры

hmdr= домашняя директория программы Необходима
prjdr= Директория проекта Необходима
velf= segy файл интервальных скоростей Файл должен иметь регулярную дискретизацию по инлайн, кросслайн и глубине. Необходим Нет значения по умолчанию
blkf= имя блок файла скоростной модели, созданного программой Необходим Нет значения по умолчанию
xldist= расстояние между кросслайнами Необходим Нет значения по умолчанию
ildist= расстояние между инлайнами Необходим Нет значения по умолчанию
vilb= номер байта для инлайна в скоростной модели Необходим Нет значения по умолчанию
vxlb= номер байта для кросслайна в скоростной модели Необходим Нет значения по умолчанию
vdz= шаг по глубине скоростной модели Необходим Нет значения по умолчанию
nddb= список узлов - Необходим Нет значения по умолчанию См: Конфигурация кластера, описание
ilincout= инкремент по инлайнам для записи по умолчанию = 5
xlincout= инкремент по кросслайнам для записи по умолчанию = 5

Список параметров в алфавитном порядке

Параметр	Описание	Необходим	Возможные значения	Значение по умолчанию
blkf=	Имя выходного блок-файла скоростей	Да		Нет
dataf=	Формат скоростной модели		0= IEEE_LE 1= IEEE_BE 2 = IBM	2 = IBM
datafo=	Формат трасс выходного файла		0= IEEE_LE 1= IEEE_BE 2 = IBM	Нет
dxmx=	максимальное значение производной по оси X			15
dymx=	максимальное значение производной по оси Y			15
dzmx=	максимальное значение производной по оси Z			50
etaf=	Имя файла с ETA-параметром			
hdrfmt=	Формат заголовков файла скоростей		0= IEEE_LE 1= IEEE_BE 2 = IBM	1= IEEE_BE
hmdr=	Директория программы			
ildist=	Расстояние между инлайнами	Да		Нет
logfile=	Имя лог-файла			
mхорху=	максимальная длина оператора вдоль осей X,Y			10
mхорз=	Максимальная длина оператора вдоль оси Z			20

nddb=	Список узлов – см. раздел <i>Конфигурация кластера</i>	Да		Нет
prjdr=	Директория проекта			Нет
resamp=	фактор передискретизации			1
twod=	2D/3D		0 = 3D 1 = 2D	0 = 3D
vdz=	Шаг дискретизации в скоростной модели	Да		Нет
velf=	Файл скоростной модели (SEG-Y). Интервалы по инлайнам, кросслайнам и глубине должны быть равномерными	Да		Нет
velfo=	имя выходного файла скоростей (SEG-Y)	да, если не создается блок-файл		нет
vilb=	Байт заголовка скоростной модели, в котором записан номер инлайна	Да		Нет
vxlb=	Байт заголовка скоростной модели, в котором записан номер кросслайна	Да		Нет
vxliltyp=	Формат скоростного файла		0 = 4 byte integer 1 = 2 byte integer	0 = 4 byte integer
xldist=	Расстояние между кросслайнами	Да		Нет

Пример файла параметров

Параметры могут располагаться в файле параметров в любом порядке. Любая строчка не являющаяся параметром автоматически становится комментарием. Все имена файлов или пары чисел заключаются в одинарные скобки. После названия параметра следует знак равенства без пробела, а значение параметра пишется через пробел. Ниже приводятся примеры задания параметров:

```

hmdr= '/data3/software/tsunami_4.17.10/smith_4.17.10'
velf= '/data1/test/grid32_vint_ibm.segy'
velfo= '/data3/smith_test/test/grid32_vint_ibm_smith-1.segy'
blkf= '/data3/smith_test/test/smooth.blk'
nddb= '/data3/smith_test/test/node.db'
prjdr= '/data3/smith_test/test'
ildist= 82.5
xldist= 82.5
resamp 3
vilb= 9
vxlb= 17
vdz= 32
dxmx= 15
dymx= 15
dzmx= 50
mxорxy= 20
mxорz= 20

```

ПРИЛОЖЕНИЕ С: ИНСТАЛЛЯЦИЯ JAVA

Для установки и использования графического интерфейса и приложения Viewer у вас на машине должна быть установлена программа Java 2 Platform, Standard Edition (J2SE) версии не ниже 1.4.2, а также Java 3D. Если у вас не установлены нужные версии, то вы (или ваш системный администратор) можете сделать это самостоятельно.

Проверьте версию Java на вашей машине:

- 1) `>java -version` выдает версию Java на вашей машине
- 2) `>which java` указывает путь к текущей версии Java.

Если команды не выдают нужную информацию, удостоверьтесь, что переменная `JAVA_HOME` наличествует в вашем файле `.cshrc` или `.bashrc`. Для правильного указания переменной обратитесь к системному администратору.

Скачивание и установка Java и Java 3D на платформах Linux и Itanium:

- 1) Зайдите на сайт Java.
<http://java.sun.com/j2se/1.5.0/download.jsp>
- 2) Нажмите кнопку **Download JDK 5.0 Update**.
- 3) Внимательно прочтите документацию по установке J2SE SDK для вашей платформы. Метод установки самораспаковывающегося бинарного файла предпочтителен.
- 4) Скачайте самораспаковывающийся бинарный файл (для вашей ОС!) в нужную директорию. 2
- 5) Следуя инструкции, установите J2SE JDK.
- 6) Вызовите текстовым редактором файл `.cshrc` (или `.bashrc`) и вставьте в него переменную `JAVA_HOME` с указанием пути к директории, в которой установлена программа.

Для `.cshrc`: `setenv JAVA_HOME директория_где_Java`

Для `.bashrc`: `export JAVA_HOME="директория_где_Java"`

- 7) К переменной `PATH` добавьте `JAVA_HOME/bin`.

Для **cshell**: переменная `PATH` находится в файле `.login`. Добавьте путь к поддиректории `bin` в директории `JAVA_HOME` в конец имеющего списка директорий при переменной `set path=` через пробел, например

```
set path=(/bin /usr/bin /sbin /usr/etc /usr/local/bin /usr/j2se_1.4.2/bin )  
где /usr/j2se_1.4.2 это ваша директория JAVA_HOME
```

Для **bourne shell**: переменная `PATH` находится в файле `.bash_profile`. Добавьте путь к поддиректории `bin` в директории `JAVA_HOME` в конец имеющего списка директорий при переменной `export PATH=` через двоеточие, например

```
export PATH=$PATH:$HOME/bin:$JAVA_HOME/bin
```

Примечание: В некоторых версиях Линукс в файле `.cshrc/.bashrc` уже есть прописанный путь к директории содержащей предустановленную версию Java. В этом случае надо вставить путь к вашей директории `$JAVA_HOME/bin` перед остальным списком после переменной `$PATH`. Таким образом, программа будет обращаться к нужной версии Java.

8) как описано в п.1, проверьте пути к Java:

1) `>java -version` выдает версию Java на вашей машине

2) `>which java` указывает путь к текущей версии Java.

9) После того, как J2SE SDK версии 1.4.2 была установлена, скачайте Java 3D SDK версии 1.3.1 и выше (остерегайтесь бета-версий) в директорию, созданную в п.5, с сайта:

<http://www.blackdown.org/java-linux/java-linux-d1.html>

Следите, чтобы программа соответствовала вашей ОС. Не забудьте скачать документацию и лицензию для Java 3D

10) После прочтения документации, установите Java 3D SDK v1.3.1 на своей машине.

11) Проверьте правильность инсталляции программы методами, указанными в документации.

Java и Java 3D на платформе SGI :

1) Удостоверьтесь, что на платформе не установлена Java SDK версии 1.4.1_06 или выше. Текущая инсталляция перепишет все предыдущие (до 1.4.1) версии Java, установленные в системе.

2) Прочтите все требования и инструкции по инсталляции на сайте SGI до скачивания нужного файла

http://www.sgi.com/products/evaluation/6.5_java2_1.4.1_06/

Выберите кнопку **Check Requirements** внизу страницы.

После прочтения требований и инструкций, выберите кнопку **Install** внизу страницы **Check Requirements**. Выберите пункт **Accept License** внизу страницы. После прочтения инструкций выберите пункт **Troubleshooter** для получения советов по инсталляции. После прочтения всех документов пройдите на одну страницу назад, к странице Installation Instructions и выберите кнопку **Install** для java2_еое.

3) После инсталляции, измените файлы users `.cshrc` или `.bashrc` включив в них путь к `JAVA_HOME`

Для `.cshrc`: `setenv JAVA_HOME path`

для `.bashrc`: `export JAVA_HOME="path"`

4) К переменной `PATH` добавьте `JAVA_HOME/bin`.

Для **cshell**: переменная `PATH` находится в файле `.login`. Добавьте путь к поддиректории `bin` в директории `JAVA_HOME` в конец имеющего списка директорий при переменной `set path=` через пробел, например

```
set path=(/bin /usr/bin /sbin /usr/etc /usr/local/bin /usr/j2se_1.4.2/bin )
где /usr/j2se_1.4.2 это ваша директория JAVA_HOME
```

Для **bourne shell**: переменная `PATH` находится в файле `.bash_profile` Добавьте путь к поддиректории `bin` в директории `JAVA_HOME` в конец имеющего списка директорий при переменной `export PATH=` через двоеточие, например

```
export PATH=$PATH:$HOME/bin:$JAVA_HOME/bin
```

5) проверьте пути к Java:

1) `>java -version` выдает версию Java на вашей машине

2) `>which java` указывает путь к текущей версии Java.

6) Скачайте и установите 3D версии 1.3.1 или выше для IRIX:

http://www.sgi.com/products/evaluation/6.5_java3d_1.3.1/

Выберите кнопку **Check Requirements** внизу страницы.

После прочтения требований и инструкций, выберите кнопку **Install** внизу страницы **Check Requirements**. Выберите пункт **Accept License** внизу страницы. После прочтения инструкций выберите пункт **Troubleshooter** для получения советов по инсталляции. После прочтения всех документов пройдите на одну страницу назад, к странице `Installation Instructions` и выберите кнопку **Install** для `java3d`.